Research Institute
Deep Reinforcement Learning

October 2022
Marketing material

Deep Reinforcement Learning, how AI helps to invest your money

An overview

Dr. Sebastian Kring
Senior Structuring Manager

Jan Keesen
Structuring Manager

IN A NUTSHELL

- In general, there is no easy path to derive financial strategies which are optimal for investor's multi-period investment objectives. Standard optimization techniques, such Markowitz optimization approach, primarily focus on single period investment objectives and are of limited use in these circumstances.
- We see AI-driven techniques such as Deep Reinforcement Learning (DRL) as the next evolutionary step to tackle multi-period optimization tasks.
- In this paper, we show how DRL can be applied to solve multi-period financial optimization tasks and, as a proof of concept, we apply to it a life cycle investing task.
- In our view, the high flexibility of the DRL approach allows us to shift the boundaries of finding optimal investment solutions in a substantial way for practitioners.

Introduction

One of the key challenges that investors face is determining financial strategies that are optimal for multi-period investment objectives. Such objectives arise naturally, and can be encountered in different investment areas, such as life cycle investing, structured fund solutions, and multi-period strategic asset allocation, among others.

In general, these multi-period optimization problems can be very difficult to solve. In trying to do so, practitioners typically apply heuristics, such as aged-based investing for life cycle investing, or Monte-Carlo simulations with different predefined financial strategies. These strategies are mostly derived from single period optimizations or historic best practices. The one with the best outcome in terms of the investor's multi-period objectives is then selected. Another - more academic - approach to solving such problems is to use "numerical dynamic programming" (see Sutton & Barto, 2018 for more on this topic). If applicable, it can identify the optimal financial strategy, but, in many cases, it cannot be applied due to the "curse of dimensionality", effectively the inescapable fact that a model's complexity grows faster than its inputs (see Taylor, 2019).

The above challenges have led us today to Deep Reinforcement Learning (DRL), which is a very promising AI-driven method for gaining key additional insights into financial strategy optimization and can supplement those already mentioned. Over the last few years, DRL has been successfully applied to multi-period financial strategy optimization, as explained in the academic literature. Two seminal papers are Buehler, Gonon, Teichman, & Wood, 2018, in which the authors presented a framework for hedging a portfolio of derivatives, and Duarte, Fonseca, Goodman, & Parker, 2021 in which optimal portfolio choices were derived in a complex lifecycle model. Both studies successfully used DRL.

The purpose of this short, introductory, paper is to demonstrate how DRL can be applied in practice. We explain the topic conceptually and discuss its advantages - mainly flexibility and generality - and also its disadvantages - explainability and local optimal solutions - and how to address them. We show readers how to implement DRL, and then apply it to a well-known, and already solved, life cycle consumption problem as a proof of concept.

This paper is organized as follows: in section one we describe what DRL is, and how it can be applied to solve multi-period financial optimization tasks, while in section two we give a short overview of different areas in asset management where multi-period financial strategy optimization is relevant, and we present an application to life cycle investing as a proof concept. Finally, section three concludes and gives an outlook.

1 / The Concept

DRL is a fast-growing research area within the field of machine learning and has witnessed several breakthroughs over the last decade or so. One of the most notable, which made headlines in the popular press, was the defeat of the world Go champion in 2016 by the computer program AlphaGo. This was a key moment in the human vs machine evolution because the extraordinary levels of complexity in the game had, until that point, been better understood by live players, and weren't simply a question of computational brute force. AlphaGo relied instead on a "reinforcement learning paradigm" effectively learning to play the game itself through repetition, and building on successful strategies while eliminating poor ones, so interacting with the environment from experience in the same way that humans do naturally.

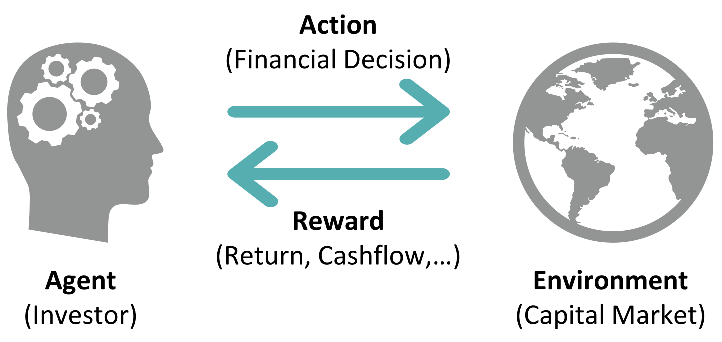
In the investment world (or, ahem, moving from a board game, to the boardroom) we can use DRL to examine how investors should make financial decisions in response to an (uncertain) capital market environment with the goal of maximizing their multi-period financial objectives. By simulating the interactions between the investor, and the capital markets, over thousands of different capital market scenarios, efficient machine learning algorithms can calibrate a neural network based financial strategy and optimize the investor's objectives.

The Model

Figure One schematically depicts the interaction model between the investor, and the capital market environment. The investor engages with the environment by making decisions, which could be investment, disinvestment, or allocation decisions. Rewards (in the form of returns or cashflows evaluated in light of the investor's investment goals) are received based on these decisions, while, simultaneously, the capital markets continue to evolve. Through these ongoing interactions, the investor learns to optimize their financial strategy by maximizing their average reward stream over thousands of different capital market scenarios. There are multiple ways to express the rewards, either through classic utility functions, risk or return measures, or via other mathematical functions that enforce certain behavior from an investor. The important point is that the goal is first specified by choice, and then maximized.

This approach relies on simulating the interaction between the "investor", represented by the neural network to be trained, and the capital market. By iteratively going through these capital market scenarios many times, the "simulated" investor learns to steadily improve their financial strategy by training the underlying neural network. We can think of it as a very specific, and very expedited, training in how best to invest.

Figure One: An interaction model between the agent and the environment in reinforcement learning



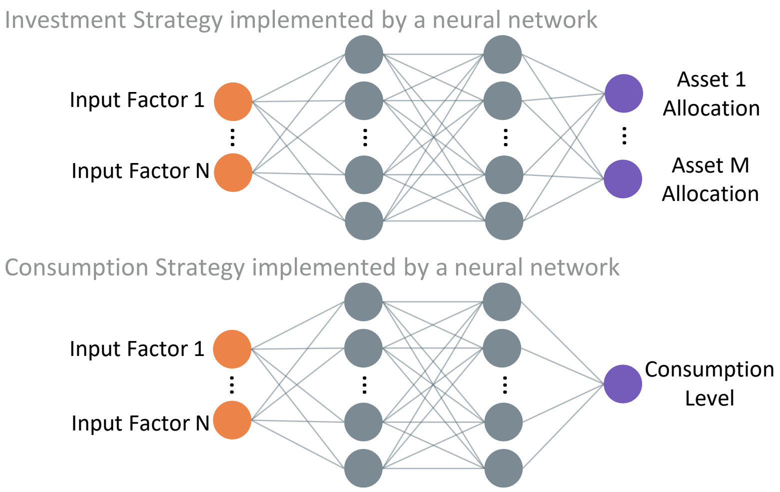
Source: DWS

In many multi-period optimization tasks, a financial strategy can be efficiently modelled by one or more neural networks. As an example, Figure Two illustrates a financial strategy that consists of two neural networks, one for the investment strategy, and the other one for the consumption strategy[1]. This figure represents a neural network that consists of one input layer

[1] In multi-period optimization problem typically the investment- and consumption strategy are highly interrelated, and the decisions of each strategy impact the other strategy. The learning process described later ensures that both strategies are optimized simultaneously capturing the effects they have on each other.

(orange nodes), two hidden layers (grey nodes) and one output layer (purple nodes). The input layer represents the information that is inserted into the decision process, which could be multiple different input factors. Two very common input factors are time, i.e., the period, and the portfolio value of the investor (see end of section two for an illustrative example). This input is processed in the hidden layers, and the output layer is the final decision. The lines between nodes represent the parameters of the networks and determine how information is passed through the network[2] (for more information about neural networks see Goodfellow, Bengio, & Courville, 2016).

Figure Two: A neural network with two hidden layers



Source: DWS

At the beginning of the learning process, the structure of the neural network(s) is determined[3], the parameters of these neural network are initialized and capital market scenarios, say 100,000 different combinations, are generated by using a capital market model. Then the first, say 256, scenarios are taken, and the financial strategy makes financial decisions, e.g., allocation- and/or consumption decisions based on these scenarios. Next, the financial results of these decisions are evaluated against the objective function – in this example 256 evaluations. Of course, these results are liable to be poor because of the random starting point (unless the best one was picked at the start by chance – good luck with that!).

The important point is that, as it now evaluates these results, the learning algorithm knows in which direction[4] it should shift the entire parameter set of the neural networks to produce - very likely - better results from the next scenarios. The next step is to analyze another 256 scenarios with these slightly shifted parameters. Then, this set of financial results are again evaluated against the objective function, and the best direction to shift the new parameters of the neural networks for the next step is also recalculated. Hopefully the idea is clear to the reader - the process is performed as an iterative loop (and, if the algorithm runs through all the original 100,000 capital market scenarios, it simply goes back to the start, taking a fresh look at the first 256 scenarios with, hopefully, a better starting point than it had before). Finally, the algorithm stops when there is no further improvement in the evaluated financial results, the financial strategy is optimized (this algorithm is known as “stochastic” or “batch gradient descent” in the literature).

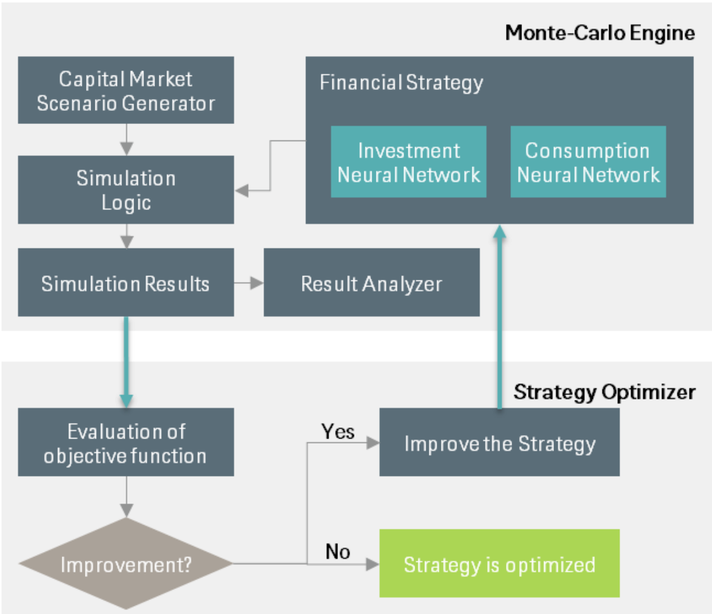
The Learning Process

Figure Three depicts the architecture of the Deep Reinforcement Learning Engine implemented by DWS. This engine performs the learning process described in the previous paragraph. The first module is the Monte-Carlo Engine. It generates

[2] Neural networks are basically functions which map inputs (input factors) to outputs. The exact mapping is determined by the parameter values of the neural network. These parameter values need to be trained in learning process.
[3] The structure of the neural networks is unchanged over the entire learning process. But the parameter values of the neural networks change in the learning process until the financial strategy is optimized.
[4] This direction is calculated as a function of the gradient of the objective function with respect to parameters of the neural networks.

the capital market scenarios according to underlying capital market models specified by the user. The second large module is called the Strategy Optimizer. This module evaluates the objective function based on the simulation results, and checks that the evaluation function has improved. Very importantly, it calculates the direction in which to shift the parameters of the neural networks in order to improve the financial strategy. Due to its modular approach the engine can be applied very flexibly to many different multi-period optimization problems by simply changing the capital market model, the financial strategy, and the objective function in the corresponding modules.

Figure Three: Architecture of the Deep Reinforcement Learning Engine



Source: DWS

The Advantages

Compared to a classic Markowitz optimization approach, the DRL learning algorithm has advantages in at least four different dimensions. Three of them are illustrated in Figure Four (which is intended as a visual indication of where, and how it can help, and not a precisely scaled diagram).

First, the DRL learning algorithm merely observes the scenarios that it's given and does not make assumptions about the specific choice of capital market model used to generate the scenarios. This is important because of its flexibility, allowing investors to use a more nuanced and comprehensive market model. For example, it is possible in the scenario generating process to allow for skewed returns, regime switches, and other enhanced features. In other words, DRL is not limited by some of the more mathematically tractable assumptions of other models.

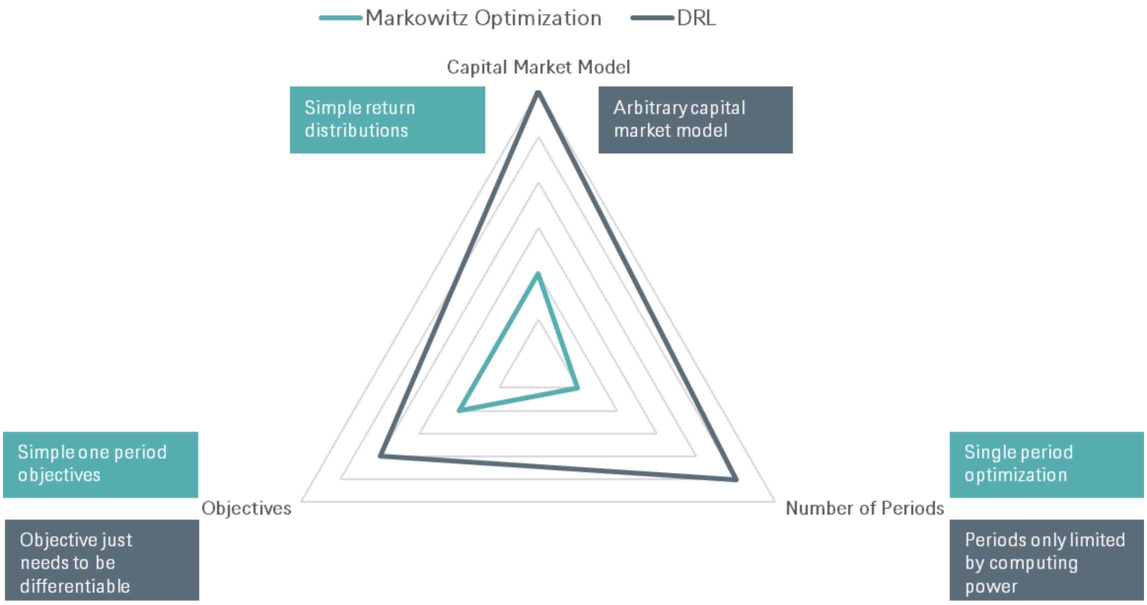
Second, there is a higher degree of flexibility in selecting the objective function. Indeed, the only requirement is that the objective function can be differentiated with respect to the parameters of the neural networks[5] (the reason – differentiation implies a gradient, and a gradient signals direction of travel for an improvement). This flexibility in choosing an objective function allows users to capture the different preferences of investors, including, for example, any behavioral preferences.

[5] This is necessary to run the algorithm. There are no guarantees on convergence.

Third, the capital market scenarios can be multi-period, such as a series of discrete quarterly returns over, say, the next twenty years for ten different asset classes per scenario. This contrasts with other approaches which evaluate only one single period, and, consequently, is clearly more practical in our view.

A final important advantage of this method is that we can generate as many scenarios as we need to calibrate the neural networks. This is a data sweet spot because one can generate both (a) as much training data as needed to solve the optimization problem, and (b) additional independent test data into analyze and test the optimized financial strategy.

Figure Four: The Advantages of DRL over Markowitz Optimization (not scaled)



Source: DWS

The Disadvantages

We are proponents of DRL in investing, and optimistic about its advantages. But it is not a panacea. There are certain disadvantages to the approach as well, and it is very important for users to know how to approach, and mitigate, them.

The first disadvantage of neural networks is their black box character. The decision-making process of an artificial neural network is hard to explain, there is no easily interpretable formula available to users that replicates its results. The second disadvantage is that the indicated financial strategy may be “overfitted”. Put simply, this means that, despite its best intentions, the approach relied too heavily on the data that it was given, and then provided a solution that was overly specific to that data. An analogy might be a soccer team that trained superbly, but only on grass, and then struggled in a match on astroturf.

To address the first challenge, the “black box” character of the financial strategy, one can study the behavior of the financial strategy, by focusing simply on different types of scenarios - adverse, average, and favorable. Depending on the intricacy of the modeled decision process, one can create tables and plots to visualize the neural networks’ mapping of inputs to outputs. Furthermore, one can analyze the key performance indicators of the strategy, e.g., risk and return measures. If an investor does not want to rely entirely on a neural network to take decisions, then they can instead derive optimized heuristics by merely using insights from the neural network based optimized financial strategy (akin to listening to what it has to say but taking it all with a pinch of salt!)

The second of the two disadvantages can be addressed by performing a new and independent simulation in the following way. First, the capital market scenario generator creates, say 10,000 scenarios which are completely independent from the ones used to train the financial strategy. Next, based on these scenarios the user conducts an ordinary simulation with the optimized financial strategy that is in question. Finally, the results of this "independent" simulation (this is the astroturf game!) are analyzed. In fact, best practice would dictate always undertaking this additional step.

In addition to the above, users can also compare the value of the objective function from the optimization procedure, with the value of the objective function from the independent out of sample simulation. If the difference between these two values is small (as one would hope), then this is a strong indicator that overfitting has not occurred (one can also supplement this analysis by checking how the key performance indicators look in the in-sample, and the out-of-sample simulation[6] - again they should be close). Together, these steps provide the necessary protection against the danger of over-fitting.

A third disadvantage of neural network optimization is that one can never be certain that the learning algorithm has found the global optimum (i.e., the best possible solution). There are two counters to this, a technical one, and a more pragmatic one. The technical answer is to note that in high dimensional optimization problems running into a local optimum is much less of a concern compared to low dimension optimization (so, put simply, the mistake of finding a solution, but not the best solution, is less likely with DRL). The reason is that there are almost always one or more dimensions left to escape a plateau[7]. Furthermore, users can leverage recent advances in machine-learning, where optimization algorithms with enhanced features, have improved the training process of neural networks dramatically. These advances are beyond the scope of this paper, but we point interested readers to the literature.

The more pragmatic solution is simply to compare the simulation results of the optimized financial strategy with ones from the rule-based strategies used as benchmarks[8]. If the financial strategy outperforms the rules-based strategies in terms of the key performance indicators - including the value of the objective function for the different strategy - then we can say that the financial strategy is the best we know for the problem at hand.

[6] An in-sample simulation uses the same paths which were applied for optimizing the financial strategy.
[7] See (Dauphin, et al., 2014) for more information.
[8] Typically, these comparisons are done by using the same capital market paths for all simulations to ensure comparability.

2 / Applications

Multi-period investment objectives arise naturally, and can be encountered in different investment areas, such as life cycle investing, structured fund solutions, and multi-period strategic asset allocation, among others. In this section we give an overview of the multi-period optimization problems in these three areas. All have typically been extremely difficult to solve with conventional techniques, and, consequently, the application of DRL has a very promising outlook.

Life cycle investing is a complex problem for most individuals. What is the optimal way to invest money given the age, wealth, risk preference and other factors of an investor? Efforts to solve this typically lead to complex multi-period financial objectives, and distinct decision processes in each period to optimize over whole the life cycle. In addition, realistic objectives might also need to capture behavioral preferences of the investor. Finally, with life cycle investing, the investor usually has to deal with both an accumulation (earn it) and a decumulation (spend it) phase. In the former the investor transforms their human capital[9] into financial assets. Shortly before retirement, the value of their financial assets peaks (they hope!) and some of the assets are then sold for cash and periodically consumed in the decumulation phase.

However, in the accumulation phase the investor not only has to make key decisions about how much to save, and how to allocate to different assets (to have sufficient financial assets for the decumulation phase), but, also, how to finance other objectives along the way, such as buying a house, financing their children's' education, or saving for retirement. In the decumulation phase the life cycle investor not only has to decide how to invest their remaining financial wealth, but also how much to withdraw from it in order to meet certain objectives such as financing vacations, ensuring a life-long consumption stream, and thinking about bequests. There are a vast number of moving parts, and, consequently, a vast literature on financial planning, life cycle investing and long-term investing[10] exists. It is precisely because of this complexity that practitioners often simply apply heuristics or use one period portfolio optimizations to advise their clients. DRL is a very promising technique to help solve these life cycle investment problems under more realistic conditions.

The second area we see as another possible application for DRL, are structured fund solutions. Investment managers whose funds have multi-period investment objectives cannot rely on a single period optimization. Examples of such funds might be ones which aim to protect a certain payment stream over multiple periods. Typically, these funds design their investment strategy by considering the current portfolio value, the remaining time to maturity, and the cashflows which need to be protected in order to derive the optimal strategy. DRL is ripe to help with this analysis.

A third area is multi-period strategic asset allocation. Institutional investors typically have multi-period investment objectives such as matching their cashflows over the next five to ten years, while minimizing certain risk metrics. An important example is funding the liabilities of a Defined-Benefit (DB) plan. The financial strategies for these plans are typically derived in an asset liability management (ALM) study. In such studies, the DB plan is simulated over the next, say, ten years by considering allocation and risk constraints predetermined by the plan sponsor. The objective of the plan sponsor is typically to minimize their shortfall risk. DRL allows plan managers to optimize their strategy directly in one simulation, without the need to apply heuristics or two-step approaches. If nothing else, it can be used as a sanity check on these approaches.

A Proof-of-Concept Example

In this final section we will solve the lifetime portfolio selection problem in discrete time, a well-known investing problem, with the DRL approach. We selected this problem not because we think that this model-framework describes the preference structure of investor fully but rather this model was solved analytically via dynamic programming in Samuelson (1969). This analytical solution allowed us to show, as a proof of concept, that the DRL approach yields the same results as in the Samuelson paper which is the first and foremost objective of this exercise in order to demonstrate the strength of the DRL approach.

[9] Human capital can be thought of as the present value of the future salary of a person.
[10] For an overview see (Campbell & Viceira, 2002).

In the lifetime selection problem, the investor has an initial wealth, of say ten thousand euros ($W_0 = €10{,}000$), which needs to be consumed over the next, say, ten periods. At the beginning of each period the investor has to make the following two decisions:

1. How much money should be withdrawn and consumed from the portfolio?
2. How should the remaining portfolio value be allocated?

The assumption is that the investor derives utility only from consumption, i.e., from withdrawing and spending money. The utility derived from this consumption is measured by a constant relative risk aversion (CRRA) utility function[11]. At the beginning of each period the investor has to make the two decisions above in order to maximize their expected cumulative CRRA utility[12] over the next ten periods.

Samuelson (1969), derived formulas for both the optimal consumption level, and the optimal allocation, providing answers to both questions. It turned out that the optimal relative consumption is independent of the portfolio value, and that the optimal allocation is independent both of time and the portfolio value. Here we replicate these results using the DRL approach.

In this example the investment universe is split into three asset classes. Equity represents a globally diversified portfolio of stocks, fixed income a globally diversified portfolio of bonds, and, finally, there is a risk-free cash component. We ignore any costs or trading frictions for simplicity. The underlying capital market is simulated as a geometric Brownian motion with the statistical parameters from Table One, and an assumed correlation of 0.12 between Equity and Fixed Income (log-returns are used and assumed to follow a normal distribution). The covariance between cash and the other two assets is assumed to be zero.

Table One: Capital market assumptions

	Expected Return	Volatility
Equity	4.6%	13.9%
Fixed Income	1.0%	6.6%
Cash	0.0%	0.0%

Source: DWS, as of as of 10/4/2022

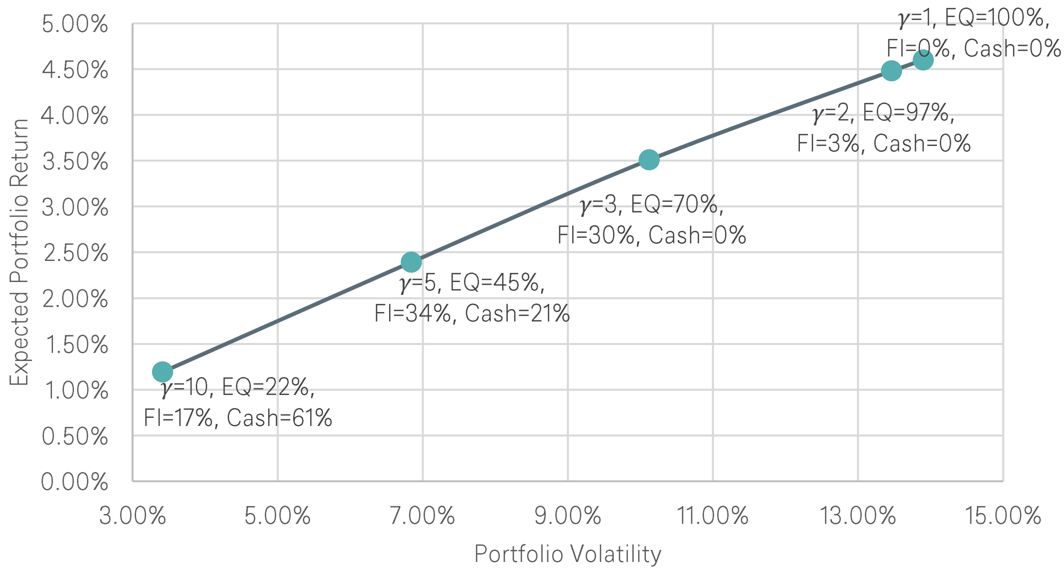
We model the investment strategy and the consumption strategy as deterministic functions of time and portfolio value, i.e., “Input Factor 1” is time, and “Input Factor 2” is portfolio value according to Figure Two. There are three outputs of the investment strategy, the relative amount that is invested into Equity, Fixed Income, and the Cash. Short selling is not allowed in this setting, and the output of the consumption network is given as the relative paid-out amount of the current portfolio value.

For our first analysis we trained neural networks for the risk aversion parameters $\gamma = 1,2,3,5,10$ and compare the initial portfolio allocation for each parameter with respect to its expected return and volatility. The resulting frontier is shown in Five, where the expected portfolio return and volatility is shown, and every point represents an optimized allocation according to the risk aversion parameter next to it. Furthermore, the equity allocation is shown right next to the point.

[11] The CRRA utility function is given by $U(c) = {}^{1}/_{1-\gamma}\, c^{1-\gamma},\ for\ \gamma > 0\ and\ \gamma \neq 1; \log(c)\ for\ \gamma = 1$, where γ is the risk aversion parameter.

[12] The following optimization problem needs to solved $max_{c_t,\ t=0,\ldots,T} E[\sum_{t=0}^{T} \delta^t\, U(c_t)]$ with the budget constraint $W_{t+1} = (1 + R_{p,t+1})(W_t - c_t)$, where c_t is the consumption at the beginning of the t-th period, $R_{p,t+1}$ is the portfolio return from t to $t+1$, δ is a discount factor, which is set to 1 in our example.

Figure Five: Hypothetical Risk Return Efficient Frontier

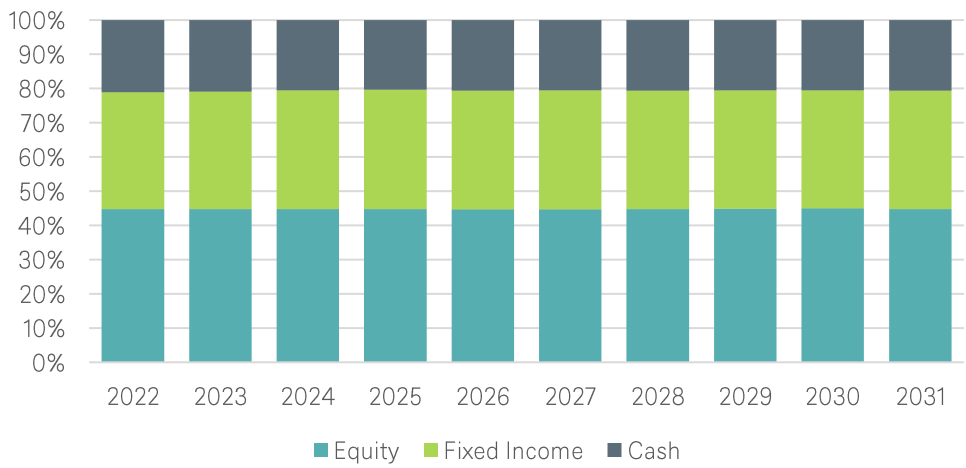


Source: DWS, as of 10/4/2022

As expected, a higher risk aversion parameter corresponds to a lower equity allocation, and therefore lower volatility of the optimized allocation, which also matches our intuitive interpretation of a risk aversion parameter. Critically, these allocations are very close to the allocation derived by Samuelson by solving the formulas in his paper.

To further analyze the fitted investment and consumption strategy, we focus on just the risk aversion parameter $\gamma = 5$. The investment allocation proposed by the neural network is dependent on the two dimensions - time and portfolio value. If we calculate the average over the second dimension in the out of sample simulation, we can plot the resulting allocation over time, which is shown in Six.

Figure Six: Average Allocation over Time



Source: DWS, as of 10/4/2022

This average allocation is approximately constant over time, which is also in line with Samuelson's findings. However, there could still be a large deviation which is dependent on the portfolio value. To explore this dimension, we calculated the standard deviation for every year in our simulation. The variation in the dimension of the portfolio value is very close to zero, which indicates that the optimal allocation is not just time-independent, but also independent of the portfolio value (as derived by Samuelson as well). This motivates a question. Why should we train a large artificial neural network to learn a complicated functional dependence if the underlying dependence is in fact rather simple? Therefore, we re-run the optimization with portfolio weights that are static over time. Table Two shows (i) the optimal allocation by applying the formulas derived by Samuelson ("Optimal"), (ii) the learned static allocation ("Static") and (iii) the mean of the learned allocation with a neural network with time and portfolio value as inputs. It can be seen that all allocations are close and differ only very slightly.

Table Two: The Learned and Optimal Allocations

	Optimal	Static	Mean allocation based on the neural network with time and portfolio value as inputs									
			2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
Equity	44.56%	44.53%	44.73%	44.75%	44.80%	44.81%	44.69%	44.64%	44.74%	44.90%	44.93%	44.73%
FI	34.35%	34.35%	34.22%	34.34%	34.70%	34.84%	34.68%	34.86%	34.68%	34.53%	34.57%	34.67%
Cash	21.09%	21.12%	21.05%	20.91%	20.50%	20.34%	20.63%	20.50%	20.58%	20.57%	20.50%	20.60%

Source: DWS, as of 10/4/2022

The additional utility gain of using the optimal allocation compared to the ones derived from the neural network with two input factors (time and portfolio value) is negligible. The relative gain is 0.02% in an out-of-sample simulation. The difference is even smaller for the allocations derived from the static case.

A similar result was observed when we analyzed the consumption strategy[13]. The relative consumption exhibits a very small standard deviation across different portfolio values but differs heavily across time. We again change the architecture of the neural network to only take time as an input. Table ThreeThree shows the consumption levels derived from the Samuelson paper ("Optimal"), the neural networks with time as the only input ("Static"), and the neural network with time and portfolio value as inputs ("Mean").

Table Three: Average Learned and Optimal Consumption

	2023	2024	2025	2026	2027	2028	2029	2030	2031
Optimal	10.43%	11.53%	12.91%	14.69%	17.06%	20.38%	25.35%	33.65%	50.23%
Static	10.42%	11.49%	12.99%	14.64%	17.05%	20.37%	25.35%	33.64%	50.23%
Mean[14]	10.43%	11.53%	12.93%	14.66%	17.08%	20.40%	25.37%	33.72%	50.28%
Standard deviation	0.00%	0.01%	0.02%	0.04%	0.06%	0.02%	0.02%	0.05%	0.17%

Source: DWS, as of 10/4/2022

Summarizing the previous observations, we see that the DRL approach finds solutions that are very close to the theoretical optimal ones derived by Samuelson. Indeed, practically speaking, these differences are negligible.

The reader then may be wondering why, if the results from both approaches are effectively the same (which is what we set out to prove), why one would favor DRL. And the answer really comes down to the ability of researchers and practitioners to build and apply DRL frameworks quite widely (Please note, theoretical solution as in the Samuelson paper are only known in rare cases). Once they are established (typically in a coding language such as R or Python), it is relatively straightforward to change the assumptions and objectives to suit the problem at hand. And this, recall, is in addition to the advantages over classical optimization techniques outlined in Figure Four.

[13] The following optimization problem needs to solved $max_{C_t,\ t=0,\dots,T} E[\sum_{t=0}^{T} \delta^t\, U(c_t)]$ with the budget constraint $W_{t+1} = (1 + R_{p,t+1})(W_t - c_t)$, where c_t is the consumption at the beginning of the t-th period, $R_{p,t+1}$ is the portfolio return from t to $t+1$, δ is a discount factor, which is set to 1 in our example.
[14] Mean consumption derived from the neural network with time and portfolio value as inputs.

3 / Conclusion and Outlook

Deep reinforcement learning is a very promising new technique for solving multi-period financial strategy optimization tasks. It also seems to be more flexible than conventional dynamic programming approaches which were historically very often used to solve such tasks. However, the "simulated" investor can only "learn" what is observed in the scenarios. Thus, a good model of the capital markets is still needed, although it can be specified to be much more realistic than that required in classical optimization.

Despite usually taking a little longer to run than traditional optimization techniques, the approach is both more flexible, and more powerful, and can be used to solve a much wider range of optimization problems. In our opinion there is huge potential to apply this technique to life cycle investing, structured fund solutions, multi-period SAAs, and, very likely, also in other currently unidentified areas. If not in isolation, then at least in conjunction with other more common approaches.

4 / Bibliography

Buehler, H., L. Gonon, J. Teichman, and B. Wood. 2018. "Deep Hedging." *arxiv: 1802.03042.*
Campbell, J. Y., and L. M. Viceira. 2002. "Strategic Asset Allocation for Long-Term Investors." *Oxford University Press.*
Dauphin, Y. N., R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. 2014. "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization." *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, (pp. 2933-2941).
Duarte, V., J. Fonseca, A. S. Goodman, and J. A. Parker. 2021. "Simple Allocation Rules and Optimal Portfolio Choice Over the Life Cycle." *Working Paper.*
Goodfellow, I., Y. Bengio, and A. Courville. 2016. "Deep Learning." *MIT Press.*
Samuelson, P. A. 1969. "Lifetime Portfolio Selection By Dynamic Stochastic Programming." *The Review of Economics and Statistics*, 239-246.
Taylor, C. R. 2019. "Dynamic Programming and the Curses of Dimensionality." *Applications Of Dynamic Programming To Agricultural Decision Problems".* CRC Press.

5 / Glossary

Deep Reinforcement Learning – A computational technique that solves problems through trial and error, adapting as it does so to prefer better solutions.

Monte-Carlo Simulations – The random sampling of thousands of possible paths from a specified statistical distribution used to recreate, and garner insights from, multiple future states of the world.

Markowitz Optimization – Per the insights of Harry Markowitz, a mathematical technique applied to investing that believes investors care about, and tried to find, the highest possible return for the lowest possible risk.

Neural Networks – A computational technique that seeks to discover meaningful patterns in complex data sets in the same way that the human brain can find them in simpler ones.

Iteration – One of a series of repetitions in a procedure or model, often with the aim of increasing the sample size, and therefore usefulness, of an outcome, or applied to a prior outcome in a process designed to improve (see Deep Reinforcement Learning)

Utility function – A mathematical attempt to quantify a person's preferences formulaically.

Algorithm – A defined set of rules that are followed in consequential order to solve a problem.

Black-box – A term used in finance to indicate that an investment methodology is opaque, typically either by design (for secrecy), or because of its complexity.

Consumption Strategy – A plan for the amount and timing of spending one's investments.

Parameters – A number, or other factor, that helps to specify or define the operation of a system or model.

Objective Function – The end goal of a process that is trying to be solved for or optimized.

Risk Aversion – The tendency, with all other factors unchanged, to prefer less risk over more.

Brownian Motion – The concept of entirely random movement (such as that of smoke through the air) originally conceived in Physics, but now regularly applied in finance as a description of, for example, certain short-term asset market returns.

Important information – EMEA, APAC & LATAM

Important information – North America

States, where such distribution, publication, availability, or use would be contrary to law or regulation or which would subject DWS to any registration or licensing requirement within such jurisdiction not currently met within such jurisdiction. Persons into whose possession this document may come are required to inform themselves of, and to observe, such restrictions.

Past performance is no guarantee of future results; nothing contained herein shall constitute any representation or warranty as to future performance. Further information is available upon investor's request. All third-party data (such as MSCI, S&P & Bloomberg) are copyrighted by and proprietary to the provider.

For Investors in Canada: No securities commission or similar authority in Canada has reviewed or in any way passed upon this document or the merits of the securities described herein and any representation to the contrary is an offence. This document is intended for discussion purposes only and does not create any legally binding obligations on the part of DWS Group. Without limitation, this document does not constitute an offer, an invitation to offer or a recommendation to enter into any transaction. When making an investment decision, you should rely solely on the final documentation relating to the transaction you are considering, and not the [document – may need to identify] contained herein. DWS Group is not acting as your financial adviser or in any other fiduciary capacity with respect to any transaction presented to you. Any transaction(s) or products(s) mentioned herein may not be appropriate for all investors and before entering into any transaction you should take steps to ensure that you fully understand such transaction(s) and have made an independent assessment of the appropriateness of the transaction(s) in the light of your own objectives and circumstances, including the possible risks and benefits of entering into such transaction. You should also consider seeking advice from your own advisers in making this assessment. If you decide to enter into a transaction with DWS Group, you do so in reliance on your own judgment. The information contained in this document is based on material we believe to be reliable; however, we do not represent that it is accurate, current, complete, or error free. Assumptions, estimates, and opinions contained in this document constitute our judgment as of the date of the document and are subject to change without notice. Any projections are based on a number of assumptions as to market conditions and there can be no guarantee that any projected results will be achieved. Past performance is not a guarantee of future results. The distribution of this document and availability of these products and services in certain jurisdictions may be restricted by law. You may not distribute this document, in whole or in part, without our express written permission.

For investors in Bermuda: This is not an offering of securities or interests in any product. Such securities may be offered or sold in Bermuda only in compliance with the provisions of the Investment Business Act of 2003 of Bermuda which regulates the sale of securities in Bermuda. Additionally, non-Bermudian persons (including companies) may not carry on or engage in any trade or business in Bermuda unless such persons are permitted to do so under applicable Bermuda legislation.

as of 10/07/22; 092393_N (10/2022)